PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06K 15/00 — A1

(11) International Publication Number: **WO 96/39680**

(43) International Publication Date: 12 December 1996 (12.12.96)

(21) International Application Number: PCT/US96/07315

(22) International Filing Date: 20 May 1996 (20.05.96)

(30) Priority Data:
08/482,877 6 June 1995 (06.06.95) US

(71) Applicant: APPLE COMPUTER, INC. [US/US]; 1 Infinite Loop - MS: 38-PAT, Cupertino, CA 95014 (US).

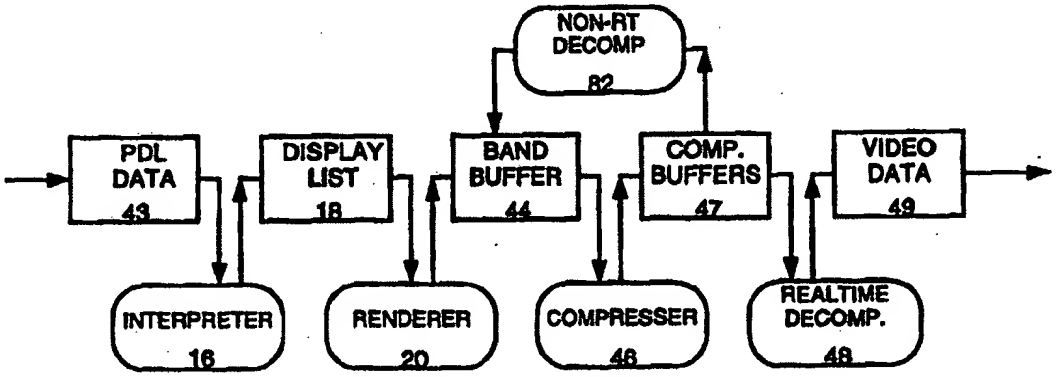
(72) Inventors: ANDRESEN, Kevin, W.; 66 Shelley Avenue, Campbell, CA 95088 (US). MOLEDINA, Riaz, A.; 3250 Woodside Road, Woodside, CA 94062 (US). SIMPSON, Mark; 22451 Franklin Court, Mountain View, CA 94040 (US). CHEN, Kok, S.; 870 E. El Camino Real #425, Sunnyvale, CA 94087 (US).

(74) Agents: CARMICHAEL, Paul, D. et al.; Apple Computer, Inc., 1 Infinite Loop - MS: 38-PAT, Cupertino, CA 95014 (US).

(81) Designated States: AL, AM, AT, AU, AZ, BB, BG, BR, BY, CA, CH, CN, CZ, DE, DK, EE, ES, FI, GB, GE, HU, IS, JP, KE, KG, KP, KR, KZ, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, TJ, TM, TR, TT, UA, UG, UZ, VN, ARIPO patent (KE, LS, MW, SD, SZ, UG), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: SYSTEM AND METHOD FOR IMAGE GENERATION USING COMPRESSION



(57) Abstract

Non-real-time decompression of stored image data permits an unlimited amount of image data to be rendered with a limited amount of available memory. When the memory available to a display list (18) is filled with image data, it is rendered into a band buffer (44) and then compressed into a compressed band buffer (47), to free up the memory used by the original display list entries. Additional entries are then entered in the display list. After the remaining entries have been captured in the display list (18), the information stored in the compressed band buffers (47) is decompressed and stored in the uncompressed band buffer (44). The additional image data in the display list is then rendered, and combined with the previously rendered data in the uncompressed band buffer (44). After the rendering is completed, the contents of the uncompressed band buffer is again compressed into the compressed band buffer format. This procedure can be continually repeated until all of the image data has been rendered into respective bands, and the page of data is complete.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AM	Armenia	GB	United Kingdom	MW	Malawi
AT	Austria	GE	Georgia	MX	Mexico
AU	Australia	GN	Guinea	NE	Niger
BB	Barbados	GR	Greece	NL	Netherlands
BE	Belgium	HU	Hungary	NO	Norway
BF	Burkina Faso	IE	Ireland	NZ	New Zealand
BG	Bulgaria	IT	Italy	PL	Poland
BJ	Benin	JP	Japan	PT	Portugal
BR	Brazil	KE	Kenya	RO	Romania
BY	Belarus	KG	Kyrgystan	RU	Russian Federation
CA	Canada	KP	Democratic People's Republic of Korea	SD	Sudan
CF	Central African Republic			SE	Sweden
CG	Congo	KR	Republic of Korea	SG	Singapore
CH	Switzerland	KZ	Kazakhstan	SI	Slovenia
CI	Côte d'Ivoire	LI	Liechtenstein	SK	Slovakia
CM	Cameroon	LK	Sri Lanka	SN	Senegal
CN	China	LR	Liberia	SZ	Swaziland
CS	Czechoslovakia	LT	Lithuania	TD	Chad
CZ	Czech Republic	LU	Luxembourg	TG	Togo
DE	Germany	LV	Latvia	TJ	Tajikistan
DK	Denmark	MC	Monaco	TT	Trinidad and Tobago
EE	Estonia	MD	Republic of Moldova	UA	Ukraine
ES	Spain	MG	Madagascar	UG	Uganda
FI	Finland	ML	Mali	US	United States of America
FR	France	MN	Mongolia	UZ	Uzbekistan
GA	Gabon	MR	Mauritania	VN	Viet Nam

SYSTEM AND METHOD FOR IMAGE GENERATION USING COMPRESSION

Technical Field

The present invention is directed to image generating systems, such as printers and video displays, and more particularly to systems of this type which employ image data compression to reduce the memory required to store image data.

Background

The spatial resolution and number of color gradients that can be produced by image generating devices such as printers and display monitors is continually increasing as technology advances. For example, commercially available color laser printers can print images at a resolution of 600 dots per inch (DPI) with 256 levels of color for each of four color components. Unfortunately, as the resolution and color depth of an image is increased, the amount of data necessary to fully describe the image also increases. For example, to store all of the data which defines a full resolution, full color image in a printer with the capabilities described above requires approximately 128 megabytes of available memory. Including this much storage capacity in a printer would make it prohibitively expensive.

To reduce the amount of memory required to store a description of an image in memory such as a page buffer or a frame buffer, data compression has been employed. As part of the image rendering process for generating data which is stored in frame buffer, the data undergoes compression. As a result, a memory having a capacity of 4-6 megabytes may be sufficient to store an entire page worth of image data. Subsequently, the data is retrieved from the frame buffer and decompressed, at a real-time rate, to be supplied to a display device to generate a video image, or a marking engine to print a page. The real-time decompression of the data can be performed by a general purpose microprocessor or a special purpose application specific integrated circuit (ASIC). Examples of general purpose processors which can be used for this purpose include the AMD29000, Motorola 68000, Intel i960 and MIPS R3000/R4000 families of microprocessors.

Of course, to take full advantage of the reduced memory requirements offered by data compression, the compression of the data must be carried out as part of the rendering process. If it were not, a full page worth of image data would first have to be rendered, and then compressed. In this approach, a full-

size page buffer would be required to store the rendered image data before it is compressed. To permit compression to be carried out in conjunction with the rendering process, and thereby avoid the need for a full-size page buffer, high level descriptions of the objects to be drawn on a page are stored in an intermediate form, known as a display list. In essence, the display list contains a series of entries which respectively represent shapes to be drawn or appearance states, e.g. color, to be applied to the objects. These entries can be stored in a high level page description language, such as PostScript. Alternatively, the display list can contain interpreted commands which are generated from the page description language. The advantage of storing the commands in a display list is the fact that the list is quite compact and describes the image at a higher level than the detailed information stored in a page buffer.

When the commands are retrieved from the display list, they are processed within the confines of non-overlapping portions of the total image. For example, a page to be printed can be divided into a number of horizontal bands. Image data pertaining to a particular band is retrieved from the display list, and rendered to produce a pixel-by-pixel representation of the image within that band. The rendered data is stored in a buffer known as a band buffer. Since a band comprises only a fraction of the total area of an image to be generated, the band buffer is much smaller in size than a complete frame buffer.

Once the data for the band has been rendered, the information in the band buffer is compressed and stored in a compressed frame buffer. The memory for the uncompressed band buffer can then be reused, to render another band of the image. The successive bands of the image are rendered and compressed, and the accumulated compressed band data in the frame buffer describes the entire image. With this approach, the memory requirements are limited to that which is sufficient to hold the display list, one uncompressed band buffer and the compressed frame buffer.

Although the number of individual objects to be drawn on a page is typically small, there is no bound to this number. As such, it can become arbitrarily large. For example, if a page of text is comprised of characters having an extremely small font size and with minimal spacing between them, a large number of characters can appear on a single page. Each character represents a different object that forms an entry on the display list. Since there is no limit on the number of objects that constitute an image, there is no bound on the size of the memory required to hold a complete display list for a

page. If the amount of memory allotted to a display list is not sufficient to store all of the image data, a display list overflow condition occurs.

In the past, the solution to this condition was to revert to a full-frame buffer printing model. In other words, no compression of the data takes place. In such a case, the remaining entries for the display list can be rendered into the frame buffer. This approach is less than satisfactory, however, since it requires sufficient memory to accommodate a full, uncompressed frame. If the available memory is not sufficient, the amount of data which defines the image must be reduced to a level which enables it to fit into the available memory. This can be accomplished, for example, by reducing the resolution of the image or the number of bits that are used to define the colors. The reduced resolution substantially degrades the image quality, and thereby fails to take advantage of the increased capabilities of modern printers.

The display list overflow condition is only one example of the situation in which it is desirable to maintain data for a complete frame in a state which permits additional data to be rendered into it. Another example of such a situation is the PostScript operator known as "copypage." This operator permits a page of data to be retained in the frame buffer after the page has been printed, so that selected portions of the data can be modified and the revised page printed. A typical application of this feature is in connection with the entry of data in a form. After the form has been rendered into the frame buffer one time, it is desirable to maintain the rendered version and change only the information that varies from one completed form to the next, e.g. names, addresses, etc. To do so, however, requires that a complete page be stored in a state which permits it to be retrieved in an uncompressed format for rendering.

It is desirable, therefore, to provide a technique which permits image data rendering to be carried out in an environment with a limited amount of available memory while not compromising the quality of the image being generated.

Summary

To address the foregoing limitations associated with prior art systems, the present invention provides a system in accordance with independent claims 1 and 12, and a method in accordance with independent claims 6 and 15. Further advantageous features, aspects and details of the invention are set forth in the dependent claims, the following description and the drawings.

The claims are to be understood as a first non-limiting approach to defining the invention in general terms.

In one aspect of the invention, non-real-time decompression is used in order to permit an unlimited amount of image data to be rendered in an environment where less than a full, uncompressed frame can be stored in available memory, and yet without degradation of the resulting image. When the memory available to the display list is filled with image data, that data is rendered into a band buffer and then compressed into a compressed band buffer, to free up the memory used by the original display list entries. Multiple compressed band buffers are accumulated to form the frame buffer. Additional entries can then be entered in the display list. After the remaining entries have been captured in the display list, or the display list is again filled, the information stored in one of the compressed band buffers is decompressed and stored in the uncompressed band buffer. The additional image data in the display list is then rendered, and combined with the previously rendered data in the uncompressed band buffer. After the rendering is completed, the contents of the uncompressed band buffer is again compressed into the compressed band buffer format. This procedure can be continually repeated until all of the image data has been rendered into the respective bands, and the page of data is complete. There is no limit to the number of display list overflow conditions and/or copypage requests that can occur, and be successfully handled, during the processing of an image.

Brief Description of the Drawings

Further features of the invention, as well as the advantages attained thereby, are explained in detail hereinafter with reference to a specific embodiment illustrated in the accompanying drawings.

Figure 1 is a block diagram of the main subsystems which make up a color printer of the type in which the present invention can be employed;

Figure 2 is a schematic illustration of the rendering process;

Figure 3 is an illustration of the division of a page into bands;

Figure 4 is a more detailed block diagram of image data flow in the implementation of the present invention;

Figure 5 is a flowchart depicting the overall process of the present invention; and

Figure 6 is a more detailed flowchart of the rendering subroutine.

Detailed Description

To facilitate an understanding of the present invention, it is described hereinafter in the context of a specific embodiment. In particular, reference is made to the implementation of the invention in a color printer which employs a multi-component color space to represent colors. It will be appreciated, however, that the practical applications of the invention are not limited to this particular embodiment. Rather, the invention can be employed in other types of image generating devices, such as CRT monitors and LCD display screens, which employ any of a variety of color spaces to represent image color.

Figure 1 is a block diagram of the major components of a color printer of a type in which the present invention can be implemented. Referring thereto, the printer 10 includes an I/O controller 12 that is connected to one or more I/O ports for communication with computers and other external sources of data to be printed. A spooler 14 accumulates image data received from the external sources, and stores the data until it is ready to be processed for printing. It will be appreciated, of course, that the spooler is optional and can be incorporated in an external device, rather than the printer itself. An interpreter 16 receives the image data and issues calls which cause the desired image to be drawn, or printed, on the paper. For example, the stream of data that is received at the I/O port may be in the form of a page description language, such as PostScript for example. In the interpreter 16, these commands are translated into various calls. The calls can be of two basic types. One set of calls identifies the appearance state of objects to be drawn. This appearance state indicates the color of the object, as well as other appearance-related factors, such as patterns or the like. The other set of calls describes the object to be drawn, such as a rectangle, a particular character of text, or the like. These calls are stored in an intermediate form, known as a display list 18, or a metafile.

The information on the display list is provided to a renderer 20. The renderer converts the object-based information from the interpreter 16 into individual pixel display values, which are stored in a frame buffer 22. The pixel display values stored in the frame buffer can undergo additional processing, such as halftone processing. Ultimately, the display values are supplied to a print engine 26, to control the actual printing of the desired image. For example, the print engine could be of the laser beam printer type. Alternatively, the print engine could be of the ink jet type.

The process which is undertaken in the renderer 20 is illustrated in greater detail in Figure 2. An exemplary document 28 to be printed on the printer contains four objects. For the sake of simplicity, these objects are represented as solid rectangles. In practice, the objects can be any geometric shape, lines, or characters of text. In the illustrated example, each object has a different color, as represented by the different shading. For example, the rectangle 30 can be cyan, the rectangle 32 can be magenta, the rectangle 34 can be yellow, and the rectangle 36 can be black.

The frame buffer 22 is comprised of a number of sections 22C, 22M, 22Y and 22K, which respectively correspond to the four color components of the color space in which the printer operates. In the illustrated embodiment, these sections are shown as four separate planes that respectively correspond to cyan, magenta, yellow and black color components. In practice, the data corresponding to the four color components can be stored in the frame buffer in any manner. For example, rather than being separated into different planes, the data for the four color components of each pixel can be stored as four successive bytes in the frame buffer.

In essence, each section of the frame buffer comprises a pixel map, with a storage location for each pixel in the image to be generated, as represented by the grid marks along the edges of each plane. In operation, the interpreter 16 issues a call to set the state of the printer to print the color cyan, and then issues a call to draw the rectangle 30. In response to the calls to draw the cyan rectangle 30, the renderer 20 stores information in the frame buffer to identify the color of each pixel in the image that is covered by the rectangle 30. The stored information includes the saturation value, or intensity, for the displayed color at the respective pixel. In a four-color system, for example, the frame buffer contains data which indicates the intensity, or amount of saturation, of each of the four color components in each pixel.

Once all of the information for the page of data is stored in the frame buffer 22, it is provided to the print engine 26, after any optional intermediate processing. More particularly, the information from each of the four frame buffer sections 22C, 22M, 22Y and 22K is individually provided to the print engine in four separate steps. For example, all of the cyan information may be printed, followed by all of the magenta information, and then the yellow and black information, to form a composite image. Without compression, the frame buffer 22 would have to be of a size sufficient to store all of the data which describes each pixel in the entire page that constitutes the image. If the printer is capable of generating 256 different intensity levels, each color

component can be represented in one byte, i.e. 8 bits of data. Thus, the four planes of the frame buffer must be able to contain a total of 32 bits of information for each pixel.

To reduce the storage requirements, the image data is compressed before being placed in the frame buffer. For this purpose, a page of data is divided into a number of non-overlapping areas. For example, as shown in Figure 3, each area can comprise a horizontal band that encompasses a portion of the image. In the example of Figure 3, the page is divided into five bands 42a-42e. In practice, the page can be divided into any number of bands. The image data is rendered separately for each band, and then compressed for storage in the frame buffer. This procedure is schematically illustrated in greater detail in Figure 4. Referring thereto, as the interpreter 16 converts the incoming stream of data from a page description language 43 to individual drawing calls, they are stored on the display list 18. Theoretically, the display list can take on any size, since there is no limit to the number of objects which a user might place in an image that is to be printed. For practical reasons, however, the amount of memory that is allotted to the display list must be confined. For example, the size of the display list might be limited to 100 kilobytes of storage.

The data stored in the display list is sorted in accordance with the bands 42a-42e into which the page is divided. The calls stored in the display list 18 for a given band of the page are retrieved by the renderer 20 and processed to generate pixel-based data, which is stored in a band buffer 44. Once all of the calls on the display list pertaining to a given band have been retrieved and rendered, the data stored in the band buffer 44 is compressed in a compressor 46. A number of different types of compression algorithms which can be employed to compress the data are known in the art, and therefore not described herein. The compressed data is stored in a compressed band buffer 47. As the image data for each band of the page is rendered and compressed, the compressed band buffers are accumulated, to form the frame buffer 22. Subsequently, during the printing of the page, the compressed data is retrieved from the frame buffer 22 and decompressed in real-time by a decompressor 48, to generate a video signal 49. This video signal is supplied to the print engine 26, to control the actual printing of the image. For example, in a laser printer, the video signal modulates the light emission from a laser diode that is scanned across a photosensitive surface.

If the number of objects which constitute an image is significant, it may be the case that the total display list is larger than that which can be accommodated in the memory allocated to the display list. In such a case, a

display list overflow condition occurs. In the present invention, this condition is accommodated by means of non-real-time decompression of the information in the frame buffer 22. This procedure is described with reference to the flowchart of Figure 5.

In operation, when a page is to be printed, a flag I is set to an initial value, e.g. zero, at Step 50. Thereafter, as each call is issued by the interpreter 16, it is captured and stored on the display list at Step 52. After each call is stored, the list is checked at Step 54 to determine whether it is full. If not, a determination is made at Step 56 whether the most recently stored call indicates the end of the page. If not, additional calls issued by the interpreter 16 are captured and stored on the display list. When the end of the page is detected at Step 56, an overflow flag is reset at Step 58, and the data on the display list is rendered at Step 60 in the normal manner.

If the image to be printed contains more objects than can be stored in the space allotted to the display list, at some point the list will be filled, which is detected at Step 54. In effect, a display list overflow condition has occurred. In response to this condition, an overflow flag is set at Step 62. The image data contained in the list is then rendered at Step 60.

After rendering, the state of the overflow flag is checked at Step 64. If the flag is not set, this is an indication that the end of the page has been reached, and the process ends. If the flag has been set, due to an overflow condition, this is an indication that there is additional image data for the page. In this case, the display list is cleared at Step 66, and the flag I is set at Step 68. Additional calls from the interpreter 16 are then captured and stored in the display list.

The subroutine for rendering the list data at Step 60 is depicted in detail in the flowchart of Figure 6. Referring thereto, when the command to render the list data is issued, the first band of the page is selected at Step 70, and the state of the flag I is checked at Step 72. If the flag is still in the reset state, this means that the end of the page has been reached or that the display list has overflowed for the first time. In this case, the image data on the display list is rendered at Step 74 and stored in the band buffer 44 in the normal fashion. After the data for the band has been rendered, it is compressed at Step 76, and stored in the appropriate compressed band buffer 47. A determination is then made at Step 78 whether there are additional bands to be rendered. If so, the next band is selected at Step 70, and the process continues in this manner, with each compressed band of data being added to the frame buffer 22. After all of

the data in the display list has been rendered and stored in the compressed band buffers, the subroutine returns to the main routine of Figure 5.

If an overflow condition has occurred, additional page data is stored on the display list to be rendered, as described above. Eventually, the command to render the list data will again be issued, either due to the end of the page or an additional overflow condition. In this situation, the flag I will have been set, which is detected at Step 72. In this situation, previously rendered data for the page is currently stored in the compressed band buffers. Therefore, the data for the band of interest is decompressed at Step 80 and stored in the band buffer 44. Thereafter, the image data currently stored in the display list is rendered into the band buffer, where it is integrated with the previously rendered data that has been retrieved from the compressed band buffer 47. Once all of the new information for the band has been retrieved from the display list and rendered, the band buffer is again compressed at Step 76.

From the foregoing, it can be appreciated that this procedure can be repeated as many times as necessary to render all of the image data for the page. In essence, the data is divided into segments, which are determined by the amount of memory allocated to the display list. The first segment of memory for a band is rendered and compressed into the compressed band buffer. When the next segment of data is stored in the display list, the previously rendered data is retrieved into the uncompressed band buffer, where the newly rendered data is integrated with it.

The decompression of the image data stored in a compressed band buffer 47 takes place in a non-real-time manner, i.e., it occurs prior to the time that the image is actually printed. Referring again to Figure 4, this decompression is separate from the real-time decompression which occurs when the video signal is being generated. If desired, the non-real-time decompression can be implemented in a device 82 that is separate from that of the real-time decompressor 48. In such a case, the two decompressors can operate in parallel with one another. More particularly, if the compressed band frame buffer is large enough to store two pages worth of data, one page can be printed using the real-time decompressor 48, while the next page is being rendered and additional data is added using the non-real-time decompressor 82. Alternatively, a single decompressor can be employed for both functions. For example, if the decompressor has sufficient bandwidth capability, it can be multiplexed between real-time and non-real-time decompression tasks.

The practical applications of the invention are not limited to the situation in which the display list overflows. Rather, it can be employed in

any situation in which new data is to be rendered after previously rendered data has been compressed. For example, it can be utilized to permit a copypage operator to be implemented, where a page with constant data can be printed multiple times and only the varying information needs to be rendered each time.

It will appreciated by those of ordinary skill in the art that the present invention can be embodied in other forms without departing from the spirit or essential characteristics thereof. For example, although disclosed in the context of a laser printer, the principles of the invention are equally applicable to other types of image generating devices which employ an intermediate form of image data storage, and data compression, such as video monitors and LCD display panels. The presently disclosed embodiments are therefore considered in all respects to be illustrative, and not restrictive. The scope of the invention is indicated by the appended claims, rather than the foregoing description, and all changes that come within the meaning and range of equivalents thereof are intended to be embraced therein.

CLAIMS

1. A system for storing compressed image data, comprising:

a first memory for storing image data in a first format, wherein said first memory has a finite storage capacity;

means for converting image data in said first memory into image data of a second format;

a second memory for storing the image data in said second format in an uncompressed form;

means for compressing the image data stored in said second memory;

a third memory for storing the compressed image data;

means for detecting an overflow condition for said first memory, wherein the amount of image data in said first format is greater than the storage capacity of said first memory;

means responsive to the detection of said overflow condition for storing additional image data in said first memory; and

a decompressor for decompressing image data stored in said third memory and storing it in said second memory in said second format for combination with additional image data that is converted from said first memory.

2. The system of Claim 1 wherein said first format comprises commands which describe objects in an image, and said second format comprises pixel data which defines image values for individual picture elements of the image.

3. The system of claim 2 wherein said pixel data has a predetermined resolution after conversion from said first format, and the decompressed data has the same resolution.

4. The system of one of Claims 1 to 3, wherein the image data is divided according to regions of an image, and wherein said second memory stores converted image data for a limited number of regions at a time, and said third memory cumulatively stores compressed image data for all of the regions.

5. The system of Claim 4 wherein said decompressor decompresses image data for one region at a time, for storage in said second memory.

6. A method for generating and storing image data, comprising the steps of:

storing image data having a first format in a first memory;

detecting that said first memory is full;

converting the image data in said first memory into a second format and storing it in a second memory in an uncompressed form;

compressing the image data in said second memory and storing it in a third memory in a compressed form;

reusing said first memory to store additional image data in said first format;

decompressing the image data stored in said third memory and storing it in said second memory in said second format in an uncompressed form; and

converting the additional image data in said first memory into said second form at and combining it with the decompressed image data stored in said second memory.

7. The method of Claim 6 wherein said first format comprises commands which describe objects in an image, and said second format comprises pixel data which defines image values for individual picture elements of the image.

8. The method of claim 7 wherein said pixel data has a predetermined resolution after conversion from said first format, and the decompressed data has the same resolution.

9. The method of one of Claims 6 to 8, further including the steps of dividing said image data according to regions of an image, and converting said image data from said first format to said second format for one region at a time.

10. The method of Claim 9 wherein said compression step is carried out for one region at a time, and further including the step of accumulating the compressed image data for the regions in said third memory.

11. The method of Claim 10 wherein said decompression step is carried out for the image data which corresponds to one region of the image.

12. A system for storing and processing data, comprising:

means for generating data to be processed;

means for processing the data;

a first memory for storing the processed data;
means for compressing the processed data stored in said first memory;
a second memory for storing the compressed data;
means for detecting the presence of additional data to be processed; and
a decompressor responsive to said detecting means for decompressing data stored in said second memory and storing it in said first memory for combination with additional data to be processed.

13. The system of claim 12, wherein said processing means comprises a renderer for converting image data from one format into another format.

14. The system of claim 12 or 13, wherein the image data is divided according to regions of an image, and wherein said first memory stores rendered image data for a limited number of regions, and said second memory cumulatively stores compressed image data for all regions of the image.

15. A method for processing and storing data, comprising the steps of:
generating data to be processed in a first format;
processing the data to convert it into a second format;
storing the converted data in a first memory;
compressing the data in said first memory and storing it in a second memory in a compressed form;
detecting that additional data in said first format is to be processed;
decompressing the data stored in said second memory and storing it in said first memory in an uncompressed form; and
converting the additional data into said second form and combining it with the decompressed data stored in said first memory.

16. The method of claim 15 further comprising the step of storing the generated data in said first format in a third memory prior to processing, and wherein said detecting step comprises detection of an overflow condition for said third memory.

17. The method of claim 15 or 16, wherein said processing step comprises image rendering wherein image data in said first format is converted into pixel data that describes individual elements of an image.

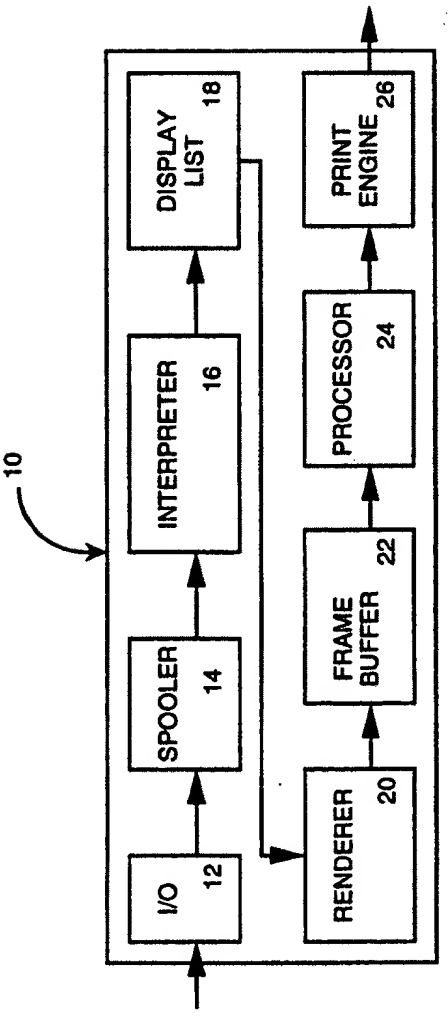

10
I/O 12
SPOOLER 14
INTERPRETER 16
DISPLAY LIST 18
RENDERER 20
FRAME BUFFER 22
PROCESSOR 24
PRINT ENGINE 26

FIG. 1

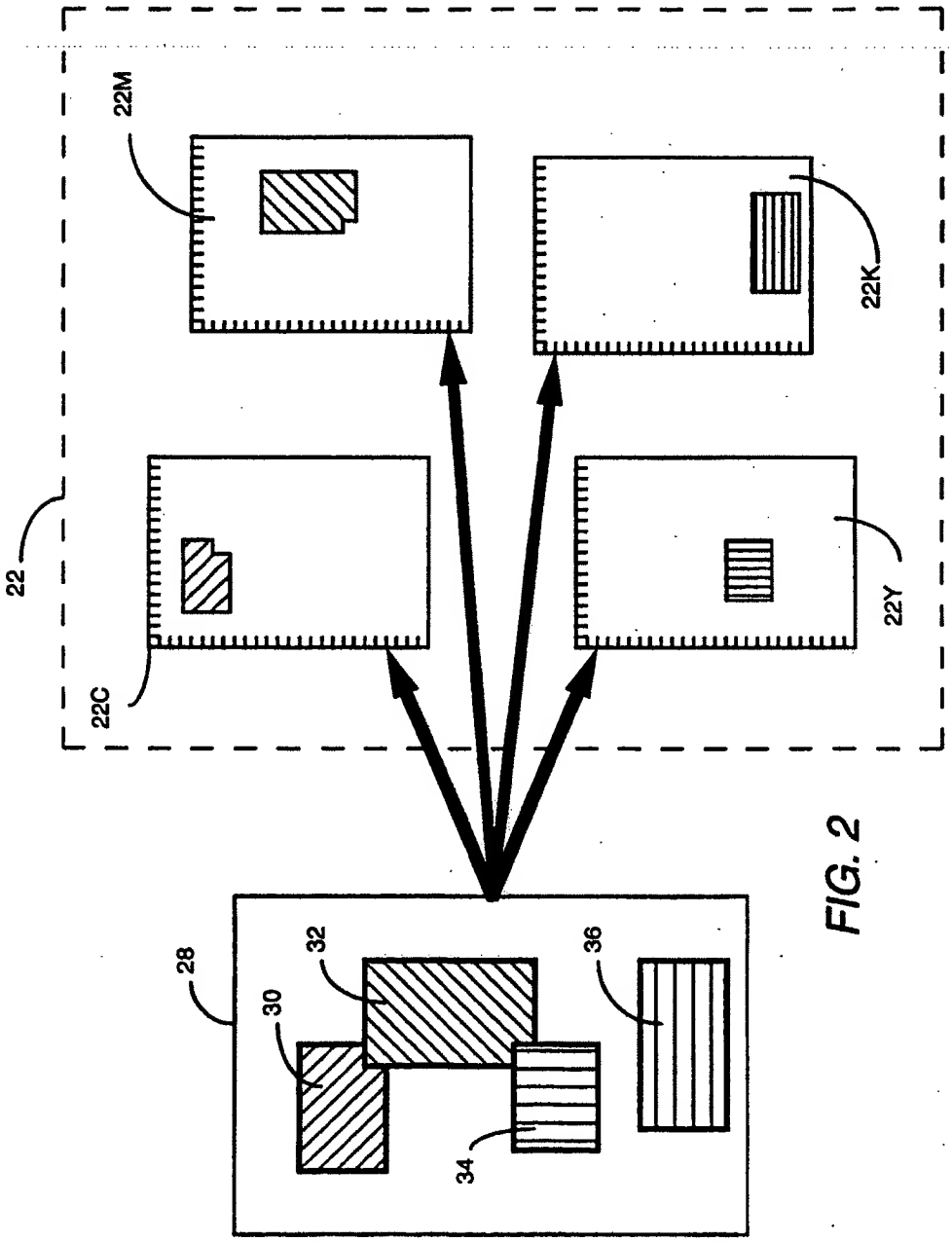


FIG. 2

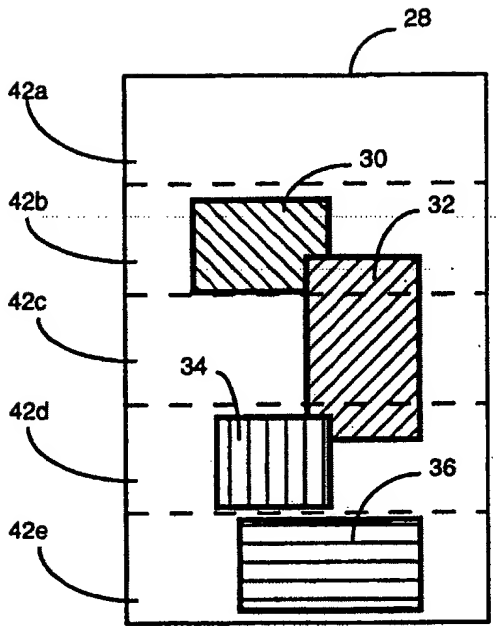

28
42a
30
42b
32
42c
34
42d
36
42e

FIG. 3

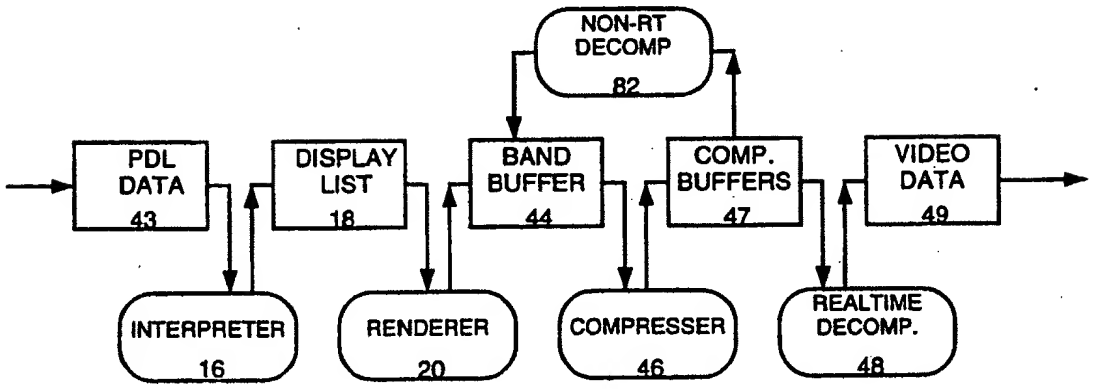

NON-RT DECOMP 82
PDL DATA 43
DISPLAY LIST 18
BAND BUFFER 44
COMP. BUFFERS 47
VIDEO DATA 49
INTERPRETER 16
RENDERER 20
COMPRESSER 46
REALTIME DECOMP. 48

FIG. 4

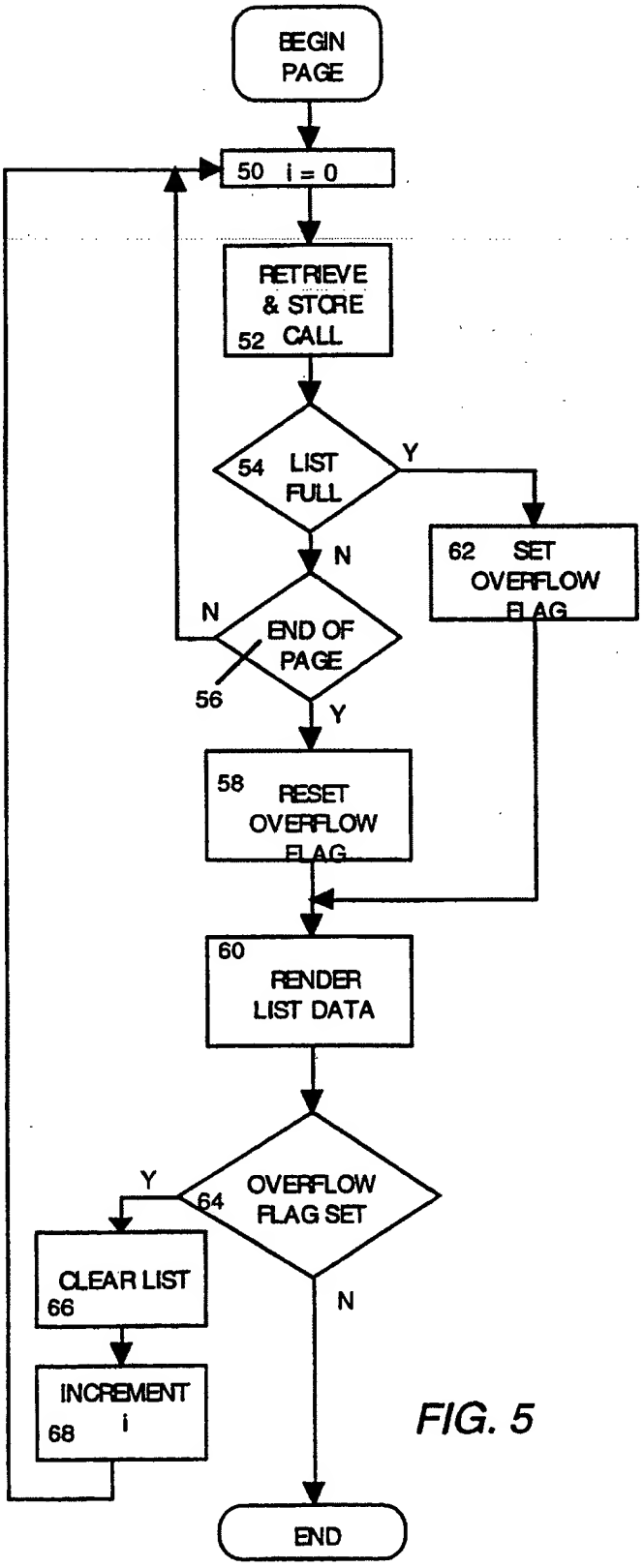

BEGIN PAGE
50 i = 0
RETRIEVE & STORE 52 CALL
54 LIST FULL
Y
N
62 SET OVERFLOW FLAG
N
END OF PAGE
56
Y
58 RESET OVERFLOW FLAG
60 RENDER LIST DATA
Y
64 OVERFLOW FLAG SET
N
CLEAR LIST 66
INCREMENT i 68
END

FIG. 5

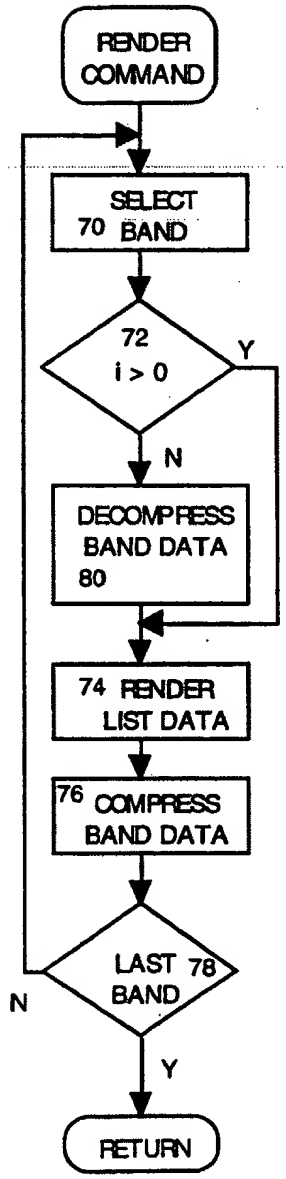

RENDER COMMAND
SELECT 70 BAND
72 i > 0
Y
N
DECOMPRESS BAND DATA 80
74 RENDER LIST DATA
76 COMPRESS BAND DATA
LAST 78 BAND
N
Y
RETURN

FIG. 6

INTERNATIONAL SEARCH REPORT

Inter nal Application No
PCT/US 96/07315

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 G06K15/00

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 6 G06K

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP,A,0 473 341 (CANON KK ;CANON INFORMATION SYST RES (AU)) 4 March 1992 see column 6, line 10 - column 8, line 35; claims; figures	1-17
X	EP,A,0 597 571 (ADOBE SYSTEMS INC) 18 May 1994 see page 5, line 33 - page 6, line 17 see page 9, line 30 - page 10, line 21; claims; figures	1-17

☐ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
9 October 1996	18.10.96

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax (+31-70) 340-3016	Gélébart, Y

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Information on patent family members

Inter nal Application No
PCT/US 96/07315

Patent document cited in search report	Publication date	Patent family member(s)		Publication date
EP-A-0473341	04-03-92	AU-B-	643053	04-11-93
		AU-A-	8177991	20-02-92
		AU-B-	640808	02-09-93
		AU-A-	8178291	20-02-92
		AU-B-	640809	02-09-93
		AU-A-	8178391	20-02-92
		EP-A-	0473340	04-03-92
		EP-A-	0475601	18-03-92
		JP-A-	7093559	07-04-95
		JP-A-	6261202	16-09-94
		JP-A-	6243243	02-09-94
		US-A-	5329616	12-07-94
EP-A-0597571	18-05-94	US-A-	5539865	23-07-96
		CA-A-	2104824	11-05-94
		JP-A-	6284297	07-10-94
		US-A-	5504842	02-04-96
		US-A-	5544290	06-08-96
		US-A-	5506944	09-04-96